**Design Document**

## User Story

URL Shortener Service like Tiny URL

**Functional Requirements:**

* Create a short URL against an original URL
* Redirect to original URL with shortened URL
* No Duplicate URL formed for same original URL.
* Expiry/long URL support

**Owner** – Manas Ranjan Sur

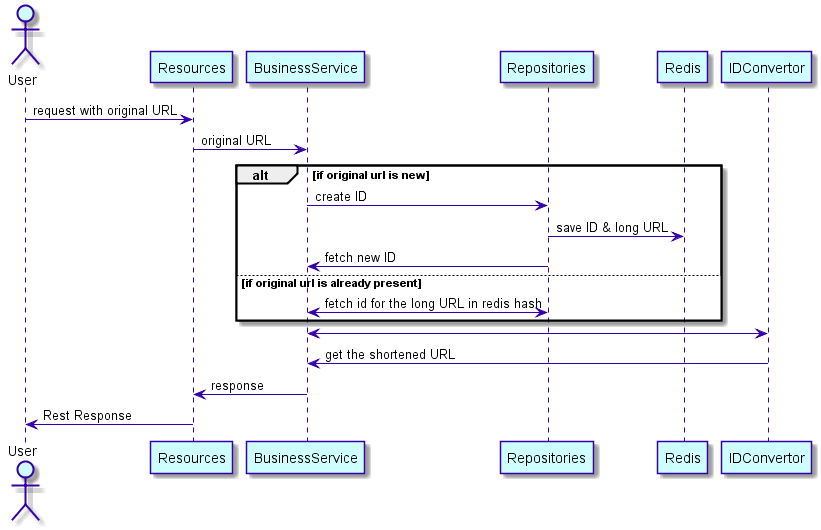
**Project Structure:**

Project contains the following packages.

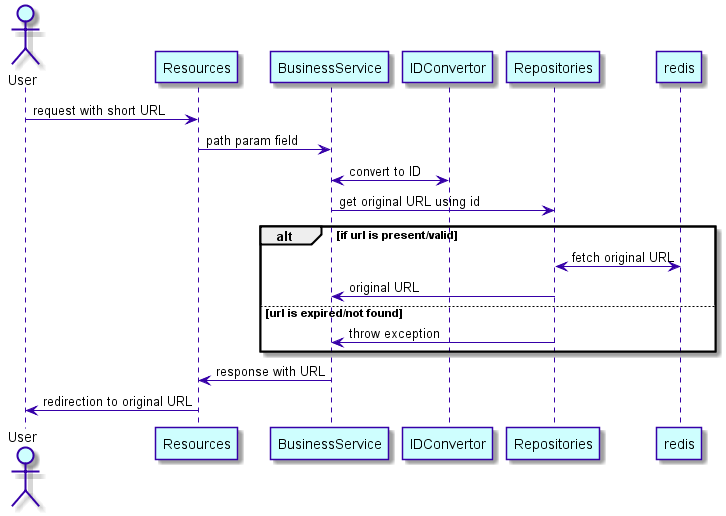
1. endpoints: All the resources will be defined here.
2. business-services All the business level functionalities are defined here.
3. repository: All data-model interaction with redis is done here.
4. datamodel: All internal entity data-types will be defined here.
5. utils: all utilities files are present in this package .

**Sequence Diagrams(using PLANTUML):**

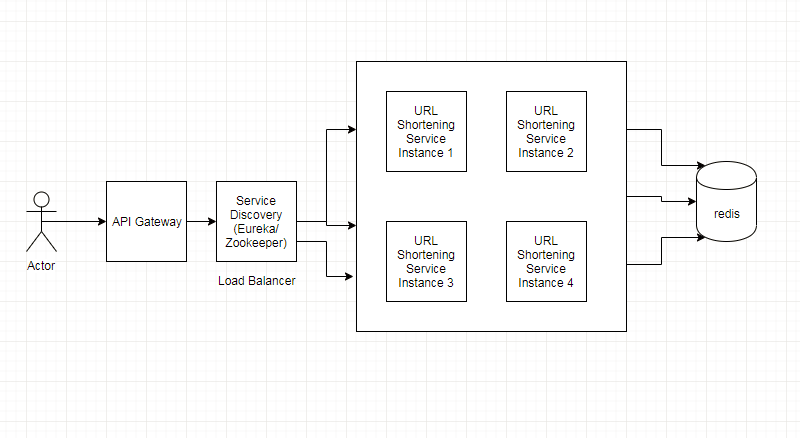
1. **UrlShorteningService**



1. **RedirectionService**

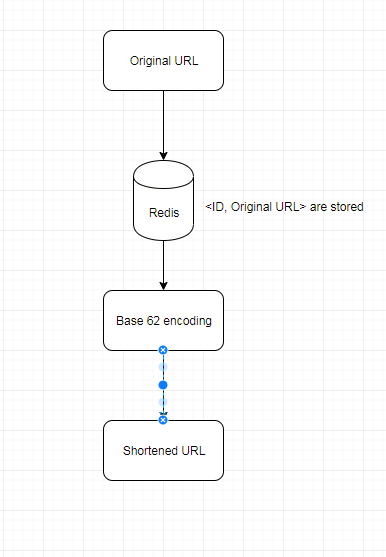


**Application Architecture:**

****

**Logic:**

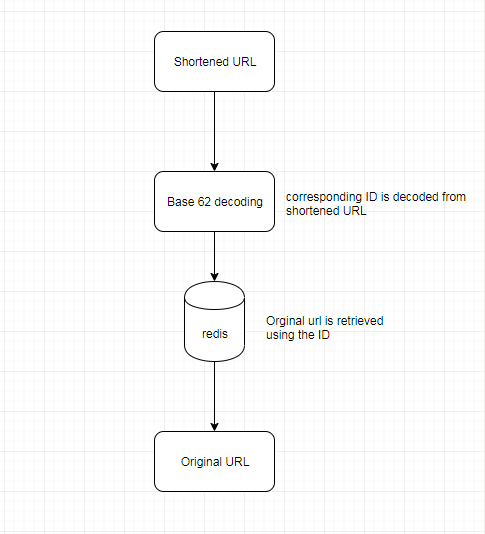
**1)Encoding**

****

Service will receive request with original string. Then a new entry will be formed in the hash(redis).If the original string is already present then it will return the same ID already present in the redis. The id will be encoded to give a shortened URL.

**Decoding**:

When someone clicks on the shortened URL, the service will convert this URL to the initial ID by base62 decoding to query the original URL from hash (redis) and redirect it to the original URL.



**Endpoints:**

**1)shorten service:** http://<domain-name>/api/app/

**Locally** : http://localhost:8008/api/app/

**Method:** post

**Input:**

**{**

**"url":"https://www.facebook.com/"**

**}**

**Output:**

**{**

**"shortUrl": "<domain-name>/api/app/i"**

**}**

1. **Redirection service :** http://<domain-name>//api/app/{id}

**Locally** : http://localhost:8008/api/app/{id}

**method: get**